--42. A computer system, comprising:

a bus;

a central processing unit coupled to said bus; and

a graphics accelerator coupled to said bus, said graphics accelerator including a texture cache system implemented as a state machine, said texture cache system including a texture cache memory that stores texels to be used by a texel value generating circuit, a replacement control component that implements a replacement policy for said texture cache memory, and a direct memory access engine that retrieves texel data from memory.

43. The computer system of claim 42, wherein said texture cache is fully associative.

44. The computer system of claim 42, wherein said replacement policy is a least recently loaded policy.

45. The computer system of claim 42, wherein said replacement policy operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

46. The computer system of claim 45, wherein said replacement policy is implemented using at least one set of flags that are associated with cache lines of said texture cache.

47. The computer system of claim 42, wherein said direct memory access engine implements a virtual-physical address translation.

48. The computer system of claim 42, wherein said texture cache system is capable of operating in a prefetch mode such that during the rendering of a first polygon, a set of texels including at least those texels needed for completely rendering a second polygon are prefetched and stored in said texture cache memory.

49. The computer system of claim 48, wherein said set of texels are prefetched if it is determined that said set of texels can fit into space available in said texture cache memory.

50. The computer system of claim 49, wherein said set of texels are prefetched if it is determined that said set of texels can fit into one half of said texture cache memory.

51. The computer system of claim 48, wherein said texture cache operates in an on demand mode for said second polygon if said set of texels cannot be prefetched.

52. A texture mapping method using a texture cache system implemented as a state machine, said texture cache system including a texture cache memory, a replacement control component, and a direct memory access engine, comprising:

(a) retrieving texels from memory via the direct memory access engine;

(b) storing said retrieved texels in the texture cache memory in accordance with a replacement policy that is determined by the replacement control component; and

(c) rendering a polygon using texels that are stored in the texture cache memory.

53. The texture mapping method of claim 52, wherein said storing comprises storing said retrieved texels in a fully associative texture cache.

54. The texture mapping method of claim 52, wherein said replacement policy is a least recently loaded policy.

55. The texture mapping method of claim 52, wherein said replacement policy operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

56. The texture mapping method of claim 55, wherein said storing comprises reviewing a state of flags within at least one set of flags that are associated with cache lines of said texture cache.

57. The texture mapping method of claim 52, wherein said retrieving comprises performing a virtual-physical address translation by the direct memory access engine.

58. The texture mapping method of claim 52, wherein said retrieving comprises prefetching texels such that during the rendering of a first polygon, a set of texels including at least those texels needed for completely rendering a second polygon are prefetched from memory.

59. The texture mapping method of claim 58, wherein said set of texels are prefetched if it is determined that said set of texels can fit into space available in said texture cache memory.

60. The texture mapping method of claim 59, wherein said set of texels are prefetched if it is determined that said set of texels can fit into one half of said texture cache memory.

61. The texture mapping method of claim 58, further comprising fetching texels on demand if said set of texels cannot be prefetched.

62. A computer system, comprising:

a memory; and

a memory control that stores two-dimensional data in said memory, wherein said data is stored in said memory using an address that is formed by interleaving individual bits of values of a coordinate in a first dimension with individual bit values of a coordinate in a second dimension.

63. The computer system of claim 62, wherein said memory and memory control are included as part of a graphics accelerator, said memory control being operative to control the storage of texels in a texture cache.

64. The computer system of claim 63, wherein said texels are stored in linear cache lines of said texture cache.

65. The computer system of claim 62, wherein said texture cache is fully associative.

66. A texture caching method, comprising:

(a) identifying a set of two-dimensional data that is to be transferred into memory; and

(b) storing said set if two-dimensional data in memory using an address that is formed by interleaving individual bits of values of a coordinate in a first dimension with individual bit values of a coordinate in a second dimension.

67. The method of claim 66, wherein said storing comprises storing texels in said memory.

68. The method of claim 67, wherein said storing comprises storing texels in memory of a graphics accelerator.